UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/727,055	12/04/2003	Kei Yoneda	56937-100	4830

7590 10/29/2008

MCDERMOTT, WILL & EMERY
600 13th Street, N.W.
Washington, DC 20005-3096

EXAMINER
BUI, HANH THI MINH

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
10/29/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/727,055	YONEDA ET AL.	
	Examiner	Art Unit	
	HANH T. BUI	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 October 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3,5,7,9,11,13,15,16,18-27 and 32-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3,5,7,9,11,13,15,16,18-27 and 32-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Status of Claims

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on October 9th, 2008 has been entered.

2. Applicants' amendment October 9th, 2008 responding to the July 9th, 2008 Final Office Action provided in the rejection of claims 1, 3, 5, 7, 9, 11, 13, 15, 16, 18-27 and 32-36.

3. Claims 37-41 have been added.

4. Claims 16, 27 and 35 have been amended.

5. Claims 1, 3, 5, 7, 9, 11, 13, 15, 16, 18-27 and 32-41 are pending in the application, of which claims 1, 16, 18, 35 and 37 are in independent form, and which have been fully considered by the examiner.

Response to Amendment

6. Any objections/rejections are not repeated below for record are withdrawn due to Applicants' amendment.

Response to Arguments

7. Applicants' arguments filed on October 9th, 2008 with respect to claims 1, 3, 5, 7, 9, 11, 13, 15, 18-26, 32-33 and 36 have been considered but are not persuasive. Therefore, the rejection of claims 1, 3, 5, 7, 9, 11, 13, 15, 18-26, 32-33 and 36 under section 103(a), which was mailed on July 9th, 2008 is maintained.

8. Applicants' arguments with respect to the newly added claims (37-41) and amended claims (16, 27, and 35) have been considered but are moot in view of the new ground(s) of rejection. See Hilla et al. (US Patent 7,155,722 – hereinafter, Hilla), Husain et al. (Pub. No. US 2003/0126260 – hereinafter, Husain), Potter (Patent No. 6,505,269 – hereinafter, Potter), and Trissel et al. (US Patent 5,274,815 - hereinafter, Trissel). It is noted that Applicants' arguments are directed towards limitations newly added via amendments, in which Husain also discloses such claimed limitations, as noted in new rejection below.

REMARKS

9. Answers To Applicant's Arguments:

a. **Argument:** However, this description does not disclose or suggest "substitut[ing] an equivalent process" that "is for the processor," as recited in claims 1 and 18 (See Remarks pp.14: 15-16).

Answer: Examiner asserts that Husain discloses in Figures 4, 9 and the associated text, such as, "… Swapping ***(substituting)*** the computer blade 401 ***(resource)*** with the computer blade 403 ***(resource)*** may involve a single computer ***switch*** from the first computer to the second computer … after the swap, the original

user ***(equivalent process)*** of the computer blade 401 will be using the computer blade 403 and the original user ***(process using the resource)*** of the computer blade 403 will be using the computer blade 401" (emphasis added – See par. [0080]) and "… if the second computer ***(resource)*** is a higher performing computer, and the user ***(equivalent process)*** of the first computer needs more computational power than the user ***(process)*** of the second computer ***(resource)***, the computers assigned to each user may be swapped ***(substitute an equivalent process for a process using the resource)*** …" (emphasis added – See par. [0108-0111]).

It is noted that the user uses applications on the computer, therefore user can be interpreted as process used in applications.

b. **Argument:** Thus, Husain's swapping is not "based upon the results of [a] determining step" "which determines the status of use of the resource," as cited in claims 1 and 18. (See Remarks pp. 15: 1-2).

Answer: Examiner asserts that Husain discloses in Figures 4, 5 and associated text, such as, "… the resource manager 409 may ***analyze the data collected*** from each computer blade in the network ***to determine one or more resource management operations*** for the computer blades. For example, a ***'resource management operation' may include operations*** for one or more of the computer blades such as, but not limited to, a single computer switch, ***a computer swap***, …" (emphasis added – See par. [0074]).

"the resource manager 409 may ***analyze data collected related to each computer*** of at least a subset of the plurality of computers in the network and ***perform***

a resource management operation based on the analysis... ***data may include*** a total memory size, a used memory size, a virtual memory size, peripheral type, available ports, processor type, processor speed, type of installed applications, whether a user is logged in, frequency of logins, ***a processor***, a hard disk, network hardware installed, ***network usage and/or status***, ***usage and/or status of installed applications***, ..." (emphasis added – See par. [0069]).

c. **Argument:** Thus, Potter does not disclose or suggest "using a same address" to "access a plurality of memory banks" – instead, different addresses are used to access the plurality of memory banks disclosed by Potter (See Remarks pp. 16: 12-14).

Answer: Examiner asserts that Potter discloses in FIG. 4 and associated text, such as, "Ext Mem 400 comprising at least one synchronous dynamic random access memory (SDRAM) array organized into a ***plurality*** of (e.g., 4) ***banks (Banks 0-3)*** ... the Ext Mem 400 comprises a plurality of memory modules or arrays 410 that is ***accessible*** by a column of processors ***(including first processor)*** ... The memory arrays logically form a single address space ***(same address)***, but they operate independently to avoid contention among the controllers ..." (emphasis added – See Col. 7: 27-40).

It should be noted that a single address space is a logical address, which is according to mapping process; a logical address can be mapped to plural physical addresses and/or plurality of memory banks. Therefore, a plurality of memory banks can be accessed using a logical address and/or same address.

d. **Argument:** For the same reasons, independent claim 35, which recites "a plurality of memory banks ... are accessible by the first processor by using the same address," is not obvious in view of the cited art (See Remarks pp. 16: 17-19).

Answer: See answer for Argument (c) above.

Claim Rejections - 35 USC § 101

10. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

11. Claims 37-41 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter, specifically directed towards computer program/software representing computer listings per se.

12. Claim 37 recites "*A compiler …comprising:*

a process identifier …

a determiner …

a substituter …"

The compiler claimed as computer program/software listings per se, i.e., the descriptions or expressions of the program, are not physical "things". They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer

element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. See Lowry, 32 F.3d at 1583-84, 32 USPQ2d at 1035. Accordingly, it is important to distinguish claims that define descriptive material per se from claims that define statutory inventions. (See MPEP 2106.01(I)).

13. Claims 38-41 do not overcome the deficiency as noted above; therefore they are also rejected as non-statutory subject matter.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 3, 5, 7, 9, 11, 13, 15 and 18-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hilla et al. (US Patent 7,155,722 – hereinafter, Hilla) in view of Husain et al. (Pub. No. US 2003/0126260 – hereinafter, Husain).

Regarding claim 1:

Hilla discloses a load balancing mechanism and technique that monitors a memory interface associated with a processor resource in a processor pool.

- *a monitoring step, which monitors a status of use of a resource identified as used by a process for processor;*

(FIG. 3 and associated text, e.g., Col. 5: 53-55; "the access monitor 500 is configured to ***monitor the activity*** (e.g., memory access requests) ***(status)*** over the memory interface ***(resource)***" - emphasis added).

- *a determining step, which determines the status of use of the resource based upon contention information obtained in the monitoring step;*

("The access monitor is arranged to compile statistics from each processor resource of the pool and provides those statistics to a central load balancing resource for use when ***determining assignment of loads*** (tasks) to the various processor resources. The inventive mechanism is arranged to differentiate between active and inactive tasks ***(contention information)*** assigned to each processor resource" (emphasis added – See Col. 3: 29-35)).

But, Hill does not explicitly teach:

- *a substituting step, which substitutes an equivalent process for a process using the resource, based upon the results of the determining step;*

- *wherein the equivalent process is for the processor, is equivalent to the process using the resource, and makes reduced use of the resource.*

However, Husain discloses in Figures 4, 5, 9 and associated text; "… Swapping ***(substituting)*** the computer blade 401 ***(resource)*** with the computer blade 403 ***(resource)*** may involve a single computer ***switch*** from the first computer to the second computer … after the swap, the original user ***(equivalent process)*** of the computer blade 401 will be using the computer blade 403 and the original user ***(process using***

the resource) of the computer blade 403 will be using the computer blade 401" (emphasis added – See par. [0080]).

"… if the second computer ***(resource)*** is a higher performing computer, and the user ***(equivalent process)*** of the first computer needs more computational power than the user ***(process)*** of the second computer ***(resource)***, the computers assigned to each user may be swapped ***(substitute an equivalent process for a process using the resource)*** …" (emphasis added – See par. [0108-0111]).

It is noted that the user uses applications on the computer, therefore user can be interpreted as process used in applications.

Husain further discloses "… the resource manager 409 may ***analyze the data collected*** from each computer blade in the network ***to determine one or more resource management operations*** for the computer blades. For example, a '***resource management operation***' ***may include operations*** for one or more of the computer blades such as, but not limited to, a single computer switch, ***a computer swap***, …" (emphasis added – See par. [0074]).

"the resource manager 409 may ***analyze data collected related to each computer*** of at least a subset of the plurality of computers in the network and ***perform a resource management operation based on the analysis***… ***data may include*** a total memory size, a used memory size, a virtual memory size, peripheral type, available ports, processor type, processor speed, type of installed applications, whether a user is logged in, frequency of logins, ***a processor***, a hard disk, network hardware

installed, ***network usage and/or status***, ***usage and/or status of installed applications***, ..." (emphasis added – See par. [0069]).

It would have been obvious to one having ordinary skill in the art at the time of the invention was made to combine the teaching of Husain into the teaching of Hilla because such combination would have provided improved systems and methods for managing resources in a system of networked computers as suggested by Husain (See para. [0009])

Regarding claim 3:

Hilla and Husain disclose *the software processing method according to claim1, wherein:*

- *the resource is a storing device for a process, and the monitoring step for the status of use monitors the status of use of the storing device.*

(FIG. 3 and associated text, e.g., Col 5: 28-29, emphasis added; "The memory 400 comprises ***storage locations addressable by the processor*** for storing software programs and data structures").

Regarding claim 5:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the monitoring step for status of use stores the preceding statuses of use of the storing device corresponding to a plurality of entries so that the contention information is generated based upon the stored status and current status of use.*

(FIG. 5 and associated text, such as, "The data structure is preferably an active session table 520 having a ***plurality of entries*** 522, each of which is associated with a corresponding session and configured to ***store status information, such as statistics, pertaining to activities associated with that session***" (emphasis added – See Col. 6: 52-56)

"the status information stored within each entry of the active session table represents ***activity (accesses)*** to the session associated with that entry within a particular time interval. During a predefined interval, each access by a processor 302 to a particular session block 420 is ***recorded*** within the active session table 520 as an increment ***(preceding statuses)*** to the status information stored within the associated entry 522. The active session table 520 thus keeps track of which sessions had a 'hit' during the interval of time, thereby indicating an active session… the central load balancer interface logic 516 provides to the central load balancer 250 a count as to the number of active sessions its associated processor 302 processed during the ***previous time*** interval" (emphasis added – See Col. 7: 35-59)).

Regarding claim 7:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the monitoring step for status of use stores the time of use when the storing device is in use, and based upon whether or not the time of use is not less than a predetermined value, the contention information is generated.*

(FIG. 5 and associated text, e.g., Col. 7: 16-34; "Over a period of time, certain active entries 522 may be ***'aged out'*** of the table 510 to allow insertion of more recent active sessions within those entries. The active session table 520 ***keeps track of the activity of sessions*** within the session region 410 of memory 400 during that time period. Upon being presented a decoded address over line 510 from the address decode logic 508, the session update logic 512 (i) updates the appropriate entry 522 (if it exists) by, e.g., incrementing status information contained within the entry or (ii) creates a new entry of the table 520. Specifically, the session update logic 512 ***compares*** that address ***(predetermined value)*** with the address of a session stored within each entry 522 of the active session table 520. If there is not a match, the session update logic 512 allocates an entry by executing an aging algorithm to remove an existing entry from the table 520 and insert the current session identified by the decoded address into that entry. The aging algorithm may be based on, e.g., a count of the number of accesses made by a processor 302 to the session block/record 420 associated with the entry" - emphasis added).

Regarding claim 9:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the resource comprises a storing device for a process and a bus that connects the processor to the storing device, and the monitoring step for status of use monitors the status of use of the bus.*

(FIG. 3 and associated text, such as, "Each resource 300 preferably comprises a processor 302 coupled to a memory 400 via a ***memory bus or interface 310***. The memory 400 comprises ***storage locations*** addressable by the processor for storing software programs and data structures" (emphasis added – See Col. 5: 25-29).

"The memory interface 310 ***(bus)*** comprises a plurality of wires or "lines," including memory address, data and control lines … ***monitors*** the physical signals on the interface 310 ***(bus)*** to determine the active/inactive status of sessions distributed among the processor resources" (emphasis added – See Col. 5: 39-40)).

Regarding claim 11:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the monitoring step for status of use stores the preceding statuses of use of the bus corresponding to a plurality of entries so that the contention information is generated based upon the stored status and current status of use.*

(FIG. 5 and associated text, such as, "The access ***monitor*** comprises memory interface logic 502 coupled to the memory interface 310 and to address decode logic 508. The memory interface logic 502 provides a physical interface to the ***memory***

interface "bus" including buffers and transceivers that receive and output information over the bus" (emphasis added – See Col. 6: 34-37)).

Regarding claim 13:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the monitoring step for status of use stores the time of use when the bus is in use, and based upon whether or not the time of use is not less than a predetermined value, the contention information is generated.*

(FIG. 5 and associated text, such as, "Over a period of time, certain active entries 522 may be '***aged out***' of the table 510 to allow insertion of more recent active sessions within those entries. The active session table 520 ***keeps track of the activity of sessions*** within the session region 410 of memory 400 during that time period. Upon being presented a decoded address over line 510 from the address decode logic 508, the session update logic 512 (i) updates the appropriate entry 522 (if it exists) by, e.g., incrementing status information contained within the entry or (ii) creates a new entry of the table 520. Specifically, the session update logic 512 ***compares*** that address ***(predetermined value)*** with the address of a session stored within each entry 522 of the active session table 520. If there is not a match, the session update logic 512 allocates an entry by executing an aging algorithm to remove an existing entry from the table 520 and insert the current session identified by the decoded address into that entry. The aging algorithm may be based on, e.g., a count of the number of accesses

made by a processor 302 to the session block/record 420 associated with the entry" (emphasis added – See Col. 7: 16-34)).

Regarding claim 15:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the resource is a second processor that executes a process in response to a processing request from the processor, and the monitoring step for status of use monitors the status of use of the second processor.*

(FIG. 2 and associated text, e.g., Col. 4: 58-60; "The intermediate network node 200 comprises a ***multiprocessor*** environment of processor resources (PS) 300 organized as a 'processor pool 210'" - emphasis added.

Examiner notes that the processor pool 210 comprises a plurality of processor resource 300, therefore the resource can be another/second processor that executes a process in response to a processing request from the processor.

FIG. 3 and associated text, e.g., Col 5: 24-29; "FIG. 3 is a schematic block diagram of a processor resource 300 ***(second processor)*** within the processor pool 210. Each resource 300 ***(second processor)*** preferably comprises a processor 302 coupled to a memory 400 via a memory bus or interface 310. The memory 400 comprises storage locations addressable by the processor for storing software programs and data structures" - emphasis added

Col. 3: 29-33; "The access ***monitor*** is arranged to compile statistics ***(status)*** from each processor resource of the pool" - emphasis added).

Regarding claim 18:

Hilla discloses a load balancing mechanism and technique that monitors a memory interface associated with a processor resource in a processor pool.

- *a monitoring step, which monitors a status of use of a resource identified as used by a process for processor;*

(FIG. 3 and the associated text, e.g., Col. 5: 53-55; "the access monitor 500 is configured to ***monitor the activity*** (e.g., memory access requests) ***(status)*** over the memory interface ***(resource)***" - emphasis added).

- *a determining step, which determines the status of use of the resource based upon contention information obtained in the monitoring step;*

(Col. 3: 29-35; "The access monitor is arranged to compile statistics from each processor resource of the pool and provides those statistics to a central load balancing resource for use when ***determining assignment of loads*** (tasks) to the various processor resources. The inventive mechanism is arranged to differentiate between active and inactive tasks ***(contention information)*** assigned to each processor resource." – emphasis added.).

- *a storing for storing contention information obtained in the monitoring step at a current time;*

(FIG. 3 and associated text, e.g., Col. 5: 53-60; "the access monitor 500 is configured to **monitor the activity** (e.g., memory access requests) ***(status)*** over the memory interface ***(resource)***. These certain regions are configured to maintain status information associated with sessions stored in the memory. By monitoring the activity associated with particular session, the access monitor 500 may gather status information, such as statistics." - emphasis added).

But, Hill does not explicitly teach:

- *a substituting step, which substitutes an equivalent process for a process using the resource, based upon the results of the determining step; wherein the equivalent process is for the processor, is equivalent to the process using the resource, and makes reduced use of the resource.*

However, Husain discloses in Figures 4, 5, 9 and associated text; "… Swapping ***(substituting)*** the computer blade 401 ***(resource)*** with the computer blade 403 ***(resource)*** may involve a single computer ***switch*** from the first computer to the second computer … after the swap, the original user ***(equivalent process)*** of the computer blade 401 will be using the computer blade 403 and the original user ***(process using the resource)*** of the computer blade 403 will be using the computer blade 401" (emphasis added – See par. [0080]).

"… if the second computer ***(resource)*** is a higher performing computer, and the user ***(equivalent process)*** of the first computer needs more computational power than the user ***(process)*** of the second computer ***(resource)***, the computers assigned to

each user may be swapped ***(substitute an equivalent process for a process using the resource)*** …" (emphasis added – See par. [0108-0111]).

It is noted that the user uses applications on the computer, therefore user can be interpreted as process used in applications.

Husain further discloses "… the resource manager 409 may ***analyze the data collected*** from each computer blade in the network ***to determine one or more resource management operations*** for the computer blades. For example, a ***' resource management operation' may include operations*** for one or more of the computer blades such as, but not limited to, a single computer switch, ***a computer swap***, …" (emphasis added – See par. [0074]) and "the resource manager 409 may ***analyze data collected related to each computer*** of at least a subset of the plurality of computers in the network and ***perform a resource management operation based on the analysis***… ***data may include*** a total memory size, a used memory size, a virtual memory size, peripheral type, available ports, processor type, processor speed, type of installed applications, whether a user is logged in, frequency of logins, ***a processor***, a hard disk, network hardware installed, ***network usage and/or status***, ***usage and/or status of installed applications***, …" (emphasis added – See par. [0069]).

It would have been obvious to one having ordinary skill in the art at the time of the invention was made to combine the teaching of Husain into the teaching of Hilla because such combination would have provided improved systems and methods for

managing resources in a system of networked computers as suggested by Husain (See para. [0009])

Regarding claim 19:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the contention information is processing time from the issuance of the processing request for the resource until the completion of the process, and the determining process for the status of use is a process which compares the processing time to a preset value.*

("the central load balancer 250 analyzes the status information provided by each access monitor 500 of a processor resource 300 to determine the actual load ***(processing time)*** executed by the resource during a specified period of time…It will be understood to those skilled in the art that various forms of status information may advantageously be gathered in accordance with the principles of the present invention. The types of statistics ***(determining process)*** that may be collected include the number of active sessions during a predetermined time interval ***(preset value)***, the number of 'hits' (accesses) to a particular session block during a time interval and the number of creations/destroys of sessions during that interval" (emphasis added – See Col. 7: 60 - Col 8: 15)).

Regarding claim 20:

The rejection of base claim 18 is incorporated. All the limitations of this claim have been noted in the rejection of claim 19.

Regarding claim 21:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the contention information is waiting time from the issuance of the processing request for the resource until the start of the process, and the determining process for the status of use is a process which compares the waiting time to a preset value.*

("It will be understood to those skilled in the art that various forms of status information ***(contention information)*** may advantageously be gathered in accordance with the principles of the present invention. The types of statistics ***(determining process)*** that may be collected include the number of active sessions during a predetermined time interval ***(preset value)***, the number of "hits" (accesses) to a particular session block during a time interval and the number of creations/destroys of sessions during that interval ***(waiting time)***" (emphasis added – See Col 8: 3-15)).

Regarding claim 22:

The rejection of base claim 18 is incorporated. All the limitations of this claim have been noted in the rejection of claim 21.

Regarding claim 23:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the determining process for status of use reexamines the determination for the status of use of the resource regularly or irregularly.*

("the central load balancer may overlay the number of active sessions per processor onto the total number of sessions assigned per processor to determine the type of activity ***(regular)*** versus inactivity ***(irregularly)*** (quiescence) across the pool of resources" (emphasis added – See Col. 7: 64-67)).

Regarding claim 24:

The rejection of base claim 18 is incorporated. All the limitations of this claim have been noted in the rejection of claim 23.

Regarding claim 25:

Hilla and Husain disclose *the software processing method according to claim 1, wherein:*

- *the determining process for status of use reexamines the determination for the status of use of the resource by using random numbers.*

(FIG. 5 and associated text, e.g., Col. 6: 37-45; "The logic 502 also provides address lines 504 and control lines 506 ***(random numbers)*** to the address decode logic 508. The address decode logic 508 is configured with conventional circuitry adapted to

decode the information presented over the lines 504, 506 in order to ***determine*** the operations performed by the processor 302 for a specific session" - emphasis added).

Regarding claim 26:

The rejection of base claim 18 is incorporated. All the limitations of this claim have been noted in the rejection of claim 25.

Regarding claim 27:

Hilla and Husain disclose *the software processing method according to claim 18 wherein:*

- *a process identifying step for identifying whether or not a process uses the resource from the software;*

- *an appearance portion identifying step for the portions of appearance of the processes identified by the process identifying step, in the case when processes to be extracted by the process identifying step are extracted from a plurality of portions of the software;*

(Hilla further discloses "… the address decode logic 508 further decodes the address information to ***identify*** a specific session block/record 420 associated with the processor request ***(whether or not a process uses the resource)***. The address decode logic 508 then instructs the session update logic 512 to (i) determine whether a session entry 522 exists for the particular decoded session and …" (emphasis added – See Col. 7: 1-15).

"each field 422 of the record 420 is software ***configurable*** (given the static nature of the record format) such that a particular field may be ***programmed*** to reflect activity used to determine balancing of sessions across the pool of processors. The hardware assist device 500 ***cooperates with the software executing on the processor*** 302 ..." (emphasis added – Col. 8: 26-36))

- *wherein the storing step stores the contention information for each of the portions of appearance so that the determining step carries out the determination by using the contention information stored for each of the portions of appearance.*

(Hilla further discloses in FIG. 3 and associated text; "The memory 400 comprises ***storage locations addressable by the processor*** for storing software programs and data structures" (emphasis added – See Col 5: 28-29).

"***each access*** by a processor ***302 to a particular session block*** 420 is recorded ***(stored)*** within the active session table 520 as an increment to the ***status information stored*** within the associated entry 522" (emphasis added – See Col. 7: 39-42)).

4. **Claim 16 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hilla in view of Potter (Patent No. 6,505,269 – hereinafter, Potter).**

Regarding claim 16:

Hilla discloses *a software processing method comprising:*

- *a monitoring step for status of use, which monitors the status of use of a second processor, the second processor performing processing in response to a processing request by a first processor;*

(FIG. 2 and associated text, e.g., Col. 4: 58-60; "The intermediate network node 200 comprises a ***multiprocessor*** environment of processor resources (PS) 300 organized as a 'processor pool 210'" – emphasis added.

Examiner notes that the processor pool 210 comprises a plurality of processor resource 300, therefore the resource can be another/second processor that executes a process in response to a processing request from the processor.

FIG. 3 and associated text, e.g., Col 5: 24-29; "FIG. 3 is a schematic block diagram of a processor resource 300 ***(second processor)*** within the processor pool 210. Each resource 300 ***(second processor)*** preferably comprises a processor 302 coupled to a memory 400 via a memory bus or interface 310. The memory 400 comprises storage locations addressable by the processor for storing software programs and data structures" - emphasis added

"The access ***monitor*** is arranged to compile statistics ***(status)*** from each processor resource of the pool" (emphasis added – See Col. 3: 29-33)).

- *an altering step for software processes, which alters software processing processes executed by the first processor or the second processor in response to contention information, the contention information being obtained in the monitoring step for status of use*

("The access monitor is arranged to compile statistics from each processor resource of the pool and provides those statistics to a central load balancing resource for use when determining assignment of loads (tasks) to the various processor resources. The inventive mechanism is arranged to differentiate between active and

inactive tasks ***(contention information)*** assigned to each processor resource. For example, a processor may have been assigned thousands of tasks, as in the case with mobile wireless traffic patterns, but the tasks or sessions may be idle ***(contention information)***. In this case, the processor may not be processing any information and, accordingly, is available to perform additional processing operations ***(altering)***. By ***monitoring*** certain memory cycles according to the inventive technique, the access monitor provides ***statistics*** to the central load balancer indicative of the ***actual measured activity of each processor resource***. The central load balancer then only needs to keep track of the total number of sessions assigned to each processor resource to guard against reaching a predefined maximum load (task) limit. Then the load balancer may assign ***(alter)*** additional tasks based upon ***(in response)*** the measured activity ***(contention information)*** received ***(obtained by)*** from the access monitor" (emphasis added – See Col. 3: 29-50)).

But, Hilla does not explicitly teach:

- *wherein the first processor can access a plurality of memory banks of a memory for the first processor by using a same address, and the plurality of memory banks includes a first memory bank and a second memory bank, the first memory bank including a program providing the processing request to the second processor, and the second memory bank including a program executed by the first processor;*

However, Potter discloses in FIG. 4 and associated text, such as, "Ext Mem 400 comprising at least one synchronous dynamic random access memory (SDRAM) array organized into a ***plurality*** of (e.g., 4) ***banks (Banks 0-3)*** … the Ext Mem 400 comprises

a plurality of memory modules or arrays 410 that is ***accessible*** by a column of processors ***(multiple processors)*** ... The memory arrays logically form a single address space ***(same address)***, but they operate independently to avoid contention among the controllers …" (emphasis added – See Col. 7: 27-40).

It should be noted that a single address space is a logical address, which is according to mapping process; a logical address can be mapped to plural physical addresses and/or plurality of memory banks. Therefore, a plurality of memory banks can be accessed using a logical address and/or same address.

- *the contention information that is a signal that indicates memory bank switching from the first memory bank to the second memory bank.*

However, Potter further discloses "this arrangement is significant to a streaming mode of operation wherein interleaving ***(switching)*** occurs between the banks ***(from the first memory bank to the second memory banks)*** and the arrays" (emphasis added – Se Col. 7: 43-45).

"... approximately 70 nsecs ***(indication of a signal)*** are required to completely cycle through a ***random access operation*** to a memory bank of the SDRAM resource operating at, e.g., 100 MHz. ***A first random access operation*** to, e.g., Bank 0 at a particular time t obviates a ***next random access*** to Bank 0 … because the bank must 'recharge' (i.e., finish the current operation and configure the storage elements for the next operation) …" (emphasis added – See Col. 7:46-60).

"the minimum time required for a random access to a location in the memory bank is approximately seven (7) cycles ***(signal)*** before another ***(switch) random access operation*** can be issue to that bank" (emphasis added – See Col. 8: 9-11).

It would have been obvious to one having ordinary skill in the art at the time of the invention was made to combine the teachings of Potter into the teachings of Hilla because such combination would have provided a technique that enables fast and efficient accesses by processors of a symmetric multiprocessor system to contiguous storage locations of a memory resource as suggested by Potter (See Col. 2: 20-22)

Regarding claim 35:

This is another system version of the rejected claim 16 above, wherein all the limitations of this claim have been noted in the rejection of claim 16.

5. Claims 32-34 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hilla in view of Potter and further in view of Trissel et al. (US Patent 5,274,815 - hereinafter, Trissel).

Regarding claim 32:

Hilla and Potter disclose *the software processing method according to claim 16, further comprising the steps of:*

- *identifying a process which uses the second processor in the software;*

(Hilla further discloses in FIG. 2 and associated text, e.g., Col. 4: lines 58-60; "The intermediate network node 200 comprises a ***multiprocessor*** environment of processor resources (PS) 300 organized as a 'processor pool 210'" - emphasis added.

Examiner notes that the processor pool 210 comprises a plurality of processor resource 300, therefore the resource can be another/second processor that executes a process in response to a processing request from the processor.)

- *mapping a process of using the first processor to the memory bank used for the first processor, and a process of using the second processor to a memory bank used for the second processor*

(Potter further discloses in FIG. 8 and associated text, e.g., Col.11: 2-14; "FIG. 8 is a schematic block diagram of the XRAM controller 800 comprising address ***mapping*** logic (i.e., ***address mapper*** 810) … The ***address mapper*** receives relevant address bits and mode select bits from the TMC processors … and converts them into interface (I/F) select, bank select, row and column bits … ***determine the particular bank and array to which the memory access operation is directed***" – emphasis added).

- *wherein the signal that indicates memory bank switching indicates switching to the memory bank used for the first processor.*

Potter further discloses "A first processor … can access a random location … at absolute time N ***(indication of a signal)***… the second processor may access the same ... location ***(same address)*** … at time N+7 ***(indication of a signal to switch)*** without contending with the first processor …" (emphasis added – See Col. 2: 45-54), "approximately 70 nsecs ***(indication of a signal)*** are required to completely cycle

through a random access operation to a memory bank of the SDRAM resource" (emphasis added – See Col. 7:46-58).

"the minimum time required for a random access to a location in the memory bank is approximately seven (7) cycles ***(signal)*** before another ***(switch)*** random access operation can be issue to that bank" (emphasis added – See Col. 8: 9-11).

But Hilla and Potter do not explicitly teach:

- *compiling a software*

However, Trissel discloses a dynamic instruction modifying controller and operation method, wherein the background teaches more clearly about the use of compiler "The most primitive way to alter a computer program is to edit the computer program, make all modifications desired by changing instructions in the computer program, ***compile***, link, and execute the code to observe changes… ***Compiler*** options are statements or commands which alter the course of the compilation process such as to determine optimization levels and enable debugging modes or facilities, and/or gather statistical information. By changing, deleting, or adding compiler options, a user or a programmer can alter the performance, execution flow, or results of a computer program. ***Compiler*** options are widely used because they are, in most cases, easier to implement than other technologies" (emphasis added - See Col. 1: 25-40).

It would have been obvious to one having ordinary skill in the art at the time of the invention was made to combine the teachings of Trissel into the teachings of Hilla and Potter because such combination would have achieved flexibility and control of software as suggested by Trissel (See Col. 1: 5-16).

Regarding claim 33:

Hilla, Potter and Trissel disclose *the software processing method according to claim 32, further comprising the step of:*

- *locking an operation of memory bank switching so that a memory bank switching cannot take place in case the first processor or the second processor is performing processing.*

Potter further discloses ”after the read (or write) command a parameter may be specified that ‘closes’ the bank ***(locking)*** …” (emphasis added – See Col. 7: 64- Col. 8: 8)

Regarding claim 34:

Hilla, Potter and Trissel disclose *the software processing method according to claim 33, further comprising the step of:*

- *unlocking the locked operation when the first processor or the second processor has finished the processing so that the signal that indicates memory bank switching can be accepted.*

Potter further discloses “In the case of random read access operation to the bank, an active command is issued … The active command functions a s a conventional read access strobe (RAS) command that ‘opens’ the bank ***(unlocking)***” (emphasis added – See Col. 7: 55-57)

Regarding claim 36:

The rejection of base claim 35 is incorporated. All the limitations of this claim have been noted in the rejection of claim 32.

6. Claims 37-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hilla in view of Husain and further in view of Trissel et al. (US Patent 5,274,815 - hereinafter, Trissel).

Regarding claim 37:

Hilla discloses *a compiler for a processor monitoring a status of use of a resource, the compiler comprising:*

- *a processor identifier that identifies a process using the resource in an input program;*

(Hilla further discloses "… the address decode logic 508 further decodes the address information to ***identify*** a specific session block/record 420 associated with the processor request ***(a process using the resource)***. The address decode logic 508 then instructs the session update logic 512 to (i) ***determine*** whether a session entry 522 exists for the particular decoded session and …" (emphasis added – See Col. 7: 1-15)).

- *a determiner that determines a status of use of the resource;*

("The access monitor is arranged to compile statistics from each processor resource of the pool and provides those statistics to a central load balancing resource for use when ***determining assignment of loads*** (tasks) to the various processor

resources. The inventive mechanism is arranged to differentiate between active and inactive tasks assigned to each processor resource" (emphasis added – See Col. 3: 29-35).

"As noted, the present invention provides a technique that efficiently monitors the physical signals on the interface 310 to ***determine*** the active/inactive ***status*** of sessions distributed among the ***processor resources***" (emphasis added – See Col. 5: 41-44)).

But, Hilla does not explicitly teach:

- *a substituter that substitutes an equivalent process which is equivalent to the process identified by the process identifier and does not use the resource for the process identified by the process identifier based upon a determination by the determiner.*

However, Husain discloses in Figures 4, 5, 9 and associated text; "… Swapping ***(substituting)*** the computer blade 401 ***(resource)*** with the computer blade 403 ***(resource)*** may involve a single computer ***switch*** from the first computer to the second computer … after the swap, the original user ***(equivalent process)*** of the computer blade 401 will be using the computer blade 403 and the original user ***(process using the resource)*** of the computer blade 403 will be using the computer blade 401" (emphasis added – See par. [0080]).

"… if the second computer ***(resource)*** is a higher performing computer, and the user ***(equivalent process)*** of the first computer needs more computational power than the user ***(process)*** of the second computer ***(resource)***, the computers assigned to

each user may be swapped ***(substitute an equivalent process for a process using the resource)*** …" (emphasis added – See par. [0108-0111]).

It should be noted that the user uses applications on the computer, therefore user can be interpreted as process used in applications. Also, there are two different users and/or processes (original user of computer blade 401 ***(equivalent process)*** and original user of computer 403 ***(identified process)***), originally the user of computer blade 410 ***(equivalent process)*** does not use the computer 403 ***(resource for the identified process)***, therefore, it meets the claimed limitation.

Husain further discloses "… the resource manager 409 may ***analyze the data collected*** from each computer blade in the network ***to determine one or more resource management operations*** for the computer blades. For example, a '***resource management operation***' ***may include operations*** for one or more of the computer blades such as, but not limited to, a single computer switch, ***a computer swap***, …" (emphasis added – See par. [0074]).

"the resource manager 409 may ***analyze data collected related to each computer*** of at least a subset of the plurality of computers in the network and ***perform a resource management operation based on the analysis***… ***data may include*** a total memory size, a used memory size, a virtual memory size, peripheral type, available ports, processor type, processor speed, type of installed applications, whether a user is logged in, frequency of logins, ***a processor***, a hard disk, network hardware installed, ***network usage and/or status***, ***usage and/or status of installed applications***, …" (emphasis added – See par. [0069]).

It would have been obvious to one having ordinary skill in the art at the time of the invention was made to combine the teaching of Husain into the teaching of Hilla because such combination would have provided improved systems and methods for managing resources in a system of networked computers as suggested by Husain (See par. [0009]).

But, Hilla and Husain do not explicitly teach

- *the compiler*

Trissel further discloses a dynamic instruction modifying controller and operation method, wherein the background teaches more clearly about the use of compiler "The most primitive way to alter a computer program is to edit the computer program, make all modifications desired by changing instructions in the computer program, ***compile***, link, and execute the code to observe changes … ***Compiler*** options are statements or commands which alter the course of the compilation process such as to determine optimization levels and enable debugging modes or facilities, and/or gather statistical information. By changing, deleting, or adding compiler options, a user or a programmer can alter the performance, execution flow, or results of a computer program. ***Compiler*** options are widely used because they are, in most cases, easier to implement than other technologies" (emphasis added - See Col. 1: 25-40).

It would have been obvious to one having ordinary skill in the art at the time of the invention was made to combine the teachings of Trissel into the teachings of Hilla and Husain because such combination would have achieved flexibility and control of software as suggested by Trissel (See Col. 1: lines15-16).

Regarding claim 38:

Hilla, Husain, and Trissel disclose *the compiler according to claim 37, wherein the determiner determines a status of use of the resource for each appearance portion of the process identified by the process identifier.*

(Hilla further discloses "… the address decode logic 508 further decodes the address information to ***identify*** a specific session block/record 420 associated with the processor request ***(whether or not a process uses the resource)***. The address decode logic 508 then instructs the session update logic 512 to (i) ***determine*** whether a session entry 522 exists for the particular decoded session and …" (emphasis added – See Col. 7: 1-15)).

Regarding claim 39:

Hilla, Husain, and Trissel disclose *the compiler according to claim 37, wherein the resource is a bus connecting the processor with a memory.*

(Hilla further discloses in FIG. 3 and associated text, such as, "Each ***resource*** 300 preferably comprises a ***processor*** 302 coupled ***(connecting)*** to a ***memory*** 400 via a ***memory bus or interface 310***. The memory 400 comprises ***storage locations*** addressable by the processor for storing software programs and data structures" (emphasis added – See Col. 5: 25-29).

"The memory interface 310 ***(bus)*** comprises a plurality of wires or "lines," including memory address, data and control lines … ***monitors*** the physical signals on

the interface 310 ***(bus)*** to determine the active/inactive status of sessions distributed among the processor resources" (emphasis added – See Col. 5: 39-40)).

Regarding claim 40:

Hilla, Husain, and Trissel disclose *the compiler according to claim 37, wherein the resource is a memory connected to the processor.*

(Hilla further discloses in FIG. 3 and associated text, such as, "Each ***resource*** 300 preferably comprises a ***processor*** 302 coupled ***(connected)*** to a ***memory*** 400 via a memory bus or interface 310. The memory 400 comprises ***storage locations*** addressable by the processor for storing software programs and data structures" (emphasis added – See Col. 5: 25-29)).

Regarding claim 41:

Hilla, Husain, and Trissel disclose *the compiler according to claim 37, wherein the resource is a second processor that performs processing in response to a request from the processor.*

(Hilla further discloses in Figure 2 and associated text, such as, "The intermediate network node 200 comprises a **multiprocessor** environment of processor resources (PS) 300 organized as a 'processor pool 210'" (emphasis added – See Col. 4: 58-60).

Examiner notes that the processor pool 210 comprises a plurality of processor resource 300, therefore the resource can be another/second processor that executes a process in response to a processing request from the processor.

FIG. 3 and associated text, such as, "FIG. 3 is a schematic block diagram of a processor resource 300 ***(second processor)*** within the processor pool 210. Each resource 300 ***(second processor)*** preferably comprises a processor 302 coupled to a memory 400 via a memory bus or interface 310. The memory 400 comprises storage locations addressable by the processor for storing software programs and data structures" (emphasis added – See Col 5: 24-29)).

Conclusion

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hanh T. Bui whose telephone number is (571) 270-1976. The examiner can normally be reached on Mon. - Thur., 9:00AM - 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/H. T. B./
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192